TP3 – Contrôle de la complexité par régularisation

Il est impératif d'avoir terminé le TP1 et le TP2 avant de passer au TP3, car ce dernier termine une série de trois séances consacrées à l'apprentissage de courbes de Bézier. Dans le TP2, vous avez estimé le degré d'une courbe de Bézier en minimisant soit l'erreur de généralisation, soit la validation croisée. On s'intéresse ici à une autre approche utilisant la notion de *régularisation*. Plus précisément, on va tenter de contrôler la complexité d'une courbe de Bézier par « estimation écrêtée » de paramètres.

Choix de paramètres adaptés à la régularisation

L'idée sous-jacente à la régularisation d'un modèle polynomial est de choisir un degré d élevé, par exemple $d = 16$, tout en évitant un sur-apprentissage (*over-fitting*) qui nuirait à la capacité de généralisation du modèle.

Dans le cadre de l'estimation de courbes de Bézier d'extrémités P_0 et P_d fixées, on modifie le paramétrage du problème de régression : au lieu d'estimer les $d-1$ ordonnées β_i des points de contrôle[1] $P_i = (\alpha_i, \beta_i)$, où $i = 1 \dots d-1$, on estime les $d-1$ écarts δ_i définis par :

$$\delta_i = \beta_i - \overline{\beta}_i \tag{1}$$

où $\overline{\beta}_i$ désigne l'ordonnée de référence, dans une situation où les points de contrôle seraient alignés (cf. figure 1). Le nouveau vecteur de paramètres δ à estimer est donc défini par :

$$\delta = [\delta_1, \dots, \delta_{d-1}] \in \mathbb{R}^{1\times(d-1)} \tag{2}$$

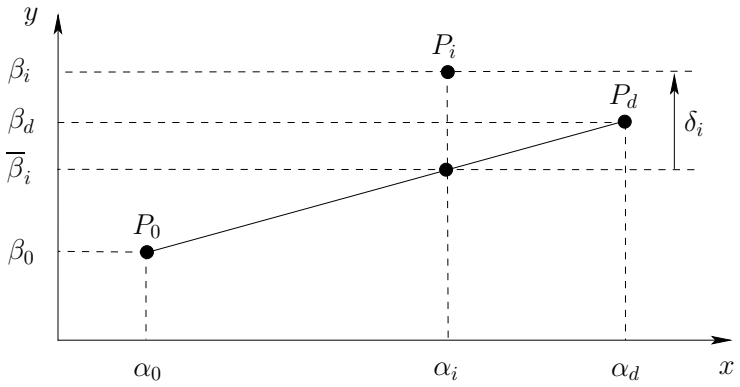

FIGURE 1 – L'écart δ_i mesure la différence entre l'ordonnée β_i du point de contrôle P_i et l'ordonnée de référence $\overline{\beta}_i$, dans une situation où les points de contrôle seraient alignés.

Dupliquez les fonctions `bezier` et `bezier_bruitee` du TP1, puis lancez le script `nouveau_parametrage`, qui superpose à la courbe de Bézier du TP1 un dessin analogue à celui de la figure 1, ce qui permet de matérialiser le vecteur de paramètres $\delta = [\delta_1, \delta_2, \delta_3, \delta_4]$.

1. Il est rappelé que les abscisses des points de contrôle sont uniformément réparties dans l'intervalle $[0, 1]$: $\alpha_i = i/d$.

Exercice 1 : estimation écrêtée des paramètres d'une courbe de Bézier

Dans le TP1, l'estimation des paramètres β du modèle, au sens des moindres carrés ordinaires, consistait à résoudre le problème suivant :

$$\widehat{\beta} = \underset{\beta=[\beta_1,\ldots,\beta_{d-1}]}{\operatorname{argmin}} \|\mathbf{A}\,\beta^\top - \mathbf{B}\|^2 \tag{3}$$

La relation (1) liant β à δ permet de réécrire le problème (3) comme suit :

$$\widehat{\delta} = \underset{\delta=[\delta_1,\ldots,\delta_{d-1}]}{\operatorname{argmin}} \|\mathbf{A}\,\delta^\top - \mathbf{C}\|^2 \tag{4}$$

où $\mathbf{C} = \mathbf{B} - \mathbf{A}\,\overline{\beta}^\top$. Pour régulariser un modèle de grande complexité, c'est-à-dire de degré d élevé, on pénalise les valeurs trop élevées des écarts δ_i. Le principe de la régression écrêtée (*ridge regression*) consiste à ajouter à l'erreur d'apprentissage un terme de pénalisation :

$$\lambda \sum_{i=1}^{d-1} {\delta_i}^2 = \lambda \|\delta\|^2 \tag{5}$$

où $\lambda \in \mathbb{R}^+$ constitue un « hyper-paramètre ». Le problème d'estimation (4) devient alors :

$$\widehat{\delta} = \underset{\delta=[\delta_1,\ldots,\delta_{d-1}]}{\operatorname{argmin}} \left\{\|\mathbf{A}\,\delta^\top - \mathbf{C}\|^2 + \lambda\,\|\delta\|^2\right\} = \underset{\delta\in\mathbb{R}^{1\times(d-1)}}{\operatorname{argmin}} \left\{(\mathbf{A}\,\delta^\top - \mathbf{C})^\top(\mathbf{A}\,\delta^\top - \mathbf{C}) + \lambda\,\delta\,\delta^\top\right\} \tag{6}$$

En notant $\mathbf{I}_{d-1}$ la matrice identité d'ordre $d-1$, on trouve aisément la solution du problème (6) :

$$\widehat{\delta}^\top = (\mathbf{A}^\top\mathbf{A} + \lambda\,\mathbf{I}_{d-1})^{-1}\mathbf{A}^\top\mathbf{C} \tag{7}$$

En vous inspirant de la fonction `moindres_carres` du TP1, écrivez la fonction `moindres_carres_ecretes`, appelée par le script `exercice_1`, qui calcule la valeur du vecteur de paramètres $\widehat{\beta}$ découlant de l'estimation (7). Attention : cette fonction doit retourner $\widehat{\beta}$ et non pas $\widehat{\delta}$.

Vérifiez que, pour $d = 16$, le choix d'une valeur très élevée de λ permet de retrouver la droite (P_0, P_d) de la figure 1. Observez l'influence de λ sur le modèle estimé.

Exercice 2 : recherche de la valeur optimale de l'hyper-paramètre λ

Dans le TP2, nous avons vu qu'en l'absence de données supplémentaires, il est possible de se servir des données d'apprentissage $\mathcal{D}_{\text{app}}$ pour tester le modèle. La validation croisée *Leave-one-out* s'écrit (cf. TP2) :

$$VC = \frac{1}{n_{\text{app}}} \sum_{j=1}^{n_{\text{app}}} \Big[y_j - f(\beta_0^*, \widehat{\beta}_j, \beta_5^*, x_j)\Big]^2 \tag{8}$$

Dans cette expression, si chaque vecteur de paramètres $\widehat{\beta}_j$ est estimé en minimisant un problème du type (7) pour les données d'apprentissage $\mathcal{D}_{\text{app}}\backslash\{(x_j, y_j)\}$, alors VC dépend non seulement du degré d de la courbe de Bézier, mais également de l'hyper-paramètre λ. Il est donc envisageable d'estimer la valeur optimale de λ en cherchant le minimum de VC, le degré de la courbe de Bézier étant fixé à une valeur élevée, par exemple $d = 16$.

Dupliquez les fonctions `calcul_VC` et `estimation_2_d_sigma` du TP2 sous les noms `calcul_VC_bis` et `estimation_lambda_sigma`. Modifiez ces deux fonctions, qui sont appelées par le script `exercice_2`, afin de montrer que la régularisation permet effectivement de contrôler la complexité du modèle (le script `exercice_2` affiche sur une même figure le modèle réel et le modèle estimé correspondant à la valeur optimale de λ). Vous constatez, en revanche, que la recherche de la valeur optimale de λ est très lente, à cause du calcul de VC.

Reformulation de la validation croisée

Vous avez étudié en cours la régression linéaire du type $y_j = \beta\,\mathbf{x}_j$, où $\beta \in \mathbb{R}^{1\times p}$ est le vecteur des paramètres du modèle et $\mathbf{x}_j \in \mathbb{R}^p$, pour j allant de 1 au nombre n_{app} de données d'apprentissage. En définissant :

$$\mathbf{Y} = \left[y_1, \ldots, y_{n_{\text{app}}}\right]^\top \in \mathbb{R}^{n_{\text{app}}} \quad \text{et} \quad \mathbf{X} = \left[\mathbf{x}_1, \ldots, \mathbf{x}_{n_{\text{app}}}\right]^\top \in \mathbb{R}^{n_{\text{app}}\times p} \tag{9}$$

la forme matricielle de la régression linéaire s'écrit :

$$\mathbf{Y} = \mathbf{X}\,\beta^\top \tag{10}$$

Si $\widehat{y}_j = \widehat{\beta}\,\mathbf{x}_j$ désigne la prédiction du modèle appris sur l'ensemble des données d'apprentissage $\mathcal{D}_{\text{app}}$, ces prédictions s'écrivent, sous forme matricielle :

$$\widehat{\mathbf{Y}} = \mathbf{H}\,\mathbf{Y} \tag{11}$$

où $\widehat{\mathbf{Y}} = \left[\widehat{y}_1, \ldots, \widehat{y}_{n_{\text{app}}}\right]^\top \in \mathbb{R}^{n_{\text{app}}}$, et où $\mathbf{H} \in \mathbb{R}^{n_{\text{app}}\times n_{\text{app}}}$, appelée « matrice chapeau », a pour expression :

$$\mathbf{H} = \mathbf{X}\,(\mathbf{X}^\top\mathbf{X})^{-1}\,\mathbf{X}^\top \tag{12}$$

Il a été vu en cours que cette matrice permet de calculer autrement la validation croisée VC :

$$VC = \frac{1}{n_{\text{app}}} \sum_{j=1}^{n_{\text{app}}} \left[\frac{y_j - \widehat{y}_j}{1 - \mathbf{H}(j,j)}\right]^2 \tag{13}$$

où $\mathbf{H}(j,j)$ désigne le $j^{\text{ème}}$ élément diagonal de $\mathbf{H}$. Or, le calcul de l'expression (13) de VC est beaucoup moins coûteux que celui de (8), dans la mesure où il ne nécessite pas d'estimer les n_{app} vecteurs de paramètres $\widehat{\beta}_j$.

Exercice 3 : nouvelle estimation de l'hyper-paramètre λ

Par identification du problème générique (10) avec l'écriture $\mathbf{A}\,\beta^\top = \mathbf{B}$ du TP1, on voit que la matrice chapeau correspondant à l'estimation des paramètres d'une courbe de Bézier s'écrit :

$$\mathbf{H} = \mathbf{A}\,(\mathbf{A}^\top\mathbf{A})^{-1}\,\mathbf{A}^\top \tag{14}$$

En lieu et place des prédictions $\widehat{\mathbf{Y}} = \mathbf{H}\,\mathbf{Y}$ utilisées dans le modèle de régression linéaire simple, avec le modèle de régression écrêtée utilisé dans ce TP, on peut prédire vectoriellement les données par une expression similaire :

$$\widehat{\mathbf{Y}} = \mathbf{S}\,\mathbf{Y} \tag{15}$$

où $\mathbf{H}$ est remplacée par la « matrice de lissage » $\mathbf{S}$ (*smoothing matrix*), dont l'écriture découle de (7) :

$$\mathbf{S} = \mathbf{A}\,(\mathbf{A}^\top\mathbf{A} + \lambda\,\mathbf{I}_{d-1})^{-1}\,\mathbf{A}^\top \tag{16}$$

L'expression de VC qui se déduit de (13) est donc :

$$VC = \frac{1}{n_{\text{app}}} \sum_{j=1}^{n_{\text{app}}} \left[\frac{y_j - f(\beta_0^*, \widehat{\beta}, \beta_5^*, x_j)}{1 - \mathbf{S}(j,j)}\right]^2 \tag{17}$$

Faites une copie de la fonction `calcul_VC_bis`, de nom `calcul_VC_ter`, que vous modifierez de manière à calculer l'expression (17) de VC. Le script `exercice_3`, qui appelle cette nouvelle fonction, doit permettre d'accélérer significativement le calcul de VC. Quelle est l'ordre de grandeur de cette accélération ?

Exercice 4 : simulation d'une flamme de bougie (question facultative)

Si vous avez réalisé la partie facultative du TP2, refaites l'estimation des modèles de silhouettes en fixant $d = 16$, et en estimant la valeur optimale de l'hyper-paramètre λ en minimisant l'expression (17) de VC.